[WEDT.A] Klasyfikacja typów serwisów WWW na postawie informacji o strukturze strony i tekstu

Michał Aniserowicz, Jakub Turek

Opis problemu

Zadanie polega na implementacji aplikacji, która dokonuje automatycznej klasyfikacji typów stron WWW na podstawie ich struktury. Analiza może obejmować źródło strony, konfigurację rozmieszczenia komponentów (layout), a także strukturę i znaczenie zamieszczonych na stronie treści.

Założenia

Projekt obejmuje implementację klasyfikatora następujących typów serwisów:

Blog rodzaj internetowego dziennika (pamiętnika), który zawiera odrębnie, chronologicznie uporządkowane wpisy. Przykład serwisu: `http://rafalstec.blox.pl/`.

Serwisy informacyjne portale zawierające najnowsze wiadomości z różnych dziedzin życia, takich jak polityka, finanse, technologie. Przykład serwisu: `http://onet.pl/`.

„Kwejki” serwisy społecznościowe oparte w głównej mierze na grafikach. Przykład serwisu: `http://kwejk.pl/`.

Sklepy internetowe portale umożliwiające zakupy w sieci przedmiotów z różnych kategorii. Przykład serwisu: `http://allegro.pl/`.

Dane wejściowe aplikacji stanowić będzie adres witryny internetowej. Na wyjście wyprowadzona zostanie nazwa kategorii lub informacja, że serwis nie został zaklasyfikowany do żadnej z powyższych kategorii.

Struktura danych

Aplikacja umożliwiać będzie budowanie pełnego drzewa HTML. W korzeniu drzewa przechowywane będą następujące informacje:

- Typ napotkanego taga HTML, na przykład `<div>`, `<h1>`.
- Dodatkowe atrybuty taga powiązane z CSS - kaskadowymi arkuszami styli: identyfikator `id="_"`, klasa `class="_"` oraz styl elementu `style="_"`.

```
<div class="tooltip-title-container">
  <div class="tooltip-title-left-corner">
    <div class="tooltip-title">
      <p class="tooltip-title-h2">
        <a href="/obrazek/1763501/autor-gry-o-tron.html">
          Autor Gry o Tron?
        </a>
      </p>
    </div>
    <div class="tooltip-title-right-corner"></div>
    <div class="clr"></div>
  </div>
</div>
```

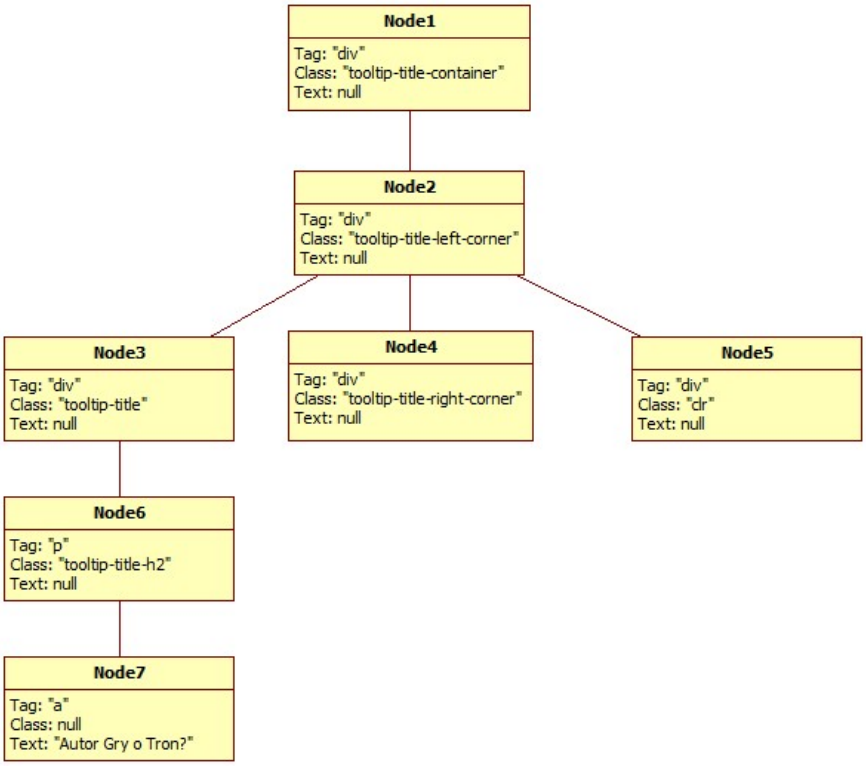
Rysunek 1: Fragment kodu źródłowego witryny http://kwejk.pl.

- Tekst zawarty pomiędzy tagiem otwierającym a zamykającym. Przykładowo dla kodu `<a>Odnośnik</a>` jest to fraza „Odnośnik”.
- Inne atrybuty kontekstowe związane z poszczególnymi tagami:
  - dla obrazka (`<img>`) - jego rozmiar oraz źródło pochodzenia (lokalne - z domeny, którą analizujemy lub zewnętrzne - spoza niej),
  - dla nagłówków (`<h1>`, `<h2>`, itd.) - rozmiar czcionki.

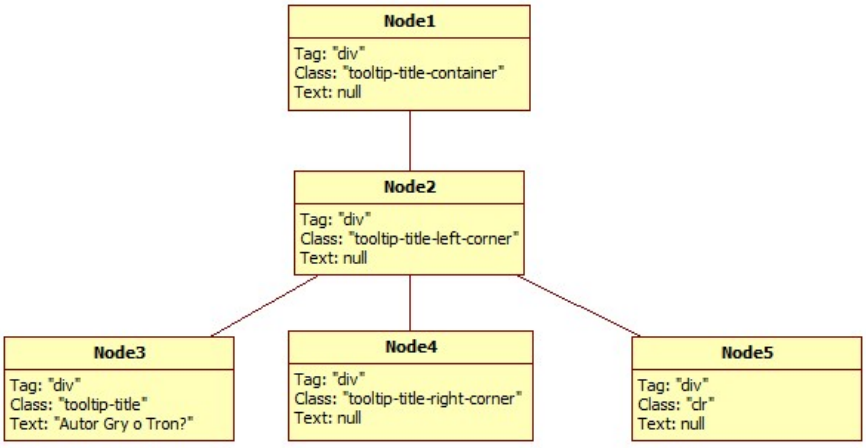
Ze względu na rozmiary oraz skomplikowanie struktury dla dużych portali, takich jak sklepy internetowe lub serwisy informacyjne, kod aplikacji będzie udostępniał różne możliwości redukcji złożoności drzewa:

- Zawężanie podzbioru tagów, dla których budowane jest drzewo. Tagi istotne dla struktury strony to, między innymi, `<div>`, `<td>`, `<article>`, `<h1>`, `<a>` oraz `<img>`. Z punktu widzenia zadania, tagi niosące niewiele informacji służą głównie do formatowania tekstu, jak na przykład `<b>`, `<span>`, oraz osadzania skryptów - `<script>`.
- Ograniczanie stopnia zagnieżdżania korzeni w drzewie:
  - pomijanie węzłów przekraczających dany, parametryzowalny, poziom zagłębienia w strukturze,
  - sklejanie kilku następujących po sobie węzłów o zbliżonych wymiarach na stronie w jeden.
- Odfiltrowywanie elementów uznanych za nieistotne metodami heurystycznymi, przykładowo prosty filtr eliminujący reklamy bazując na klasach obiektów.

Algorytm



Rysunek 2: Pełne drzewo HTML dla kodu przedstawionego na listingu 1.



Rysunek 3: Drzewo HTML z rysunku 2 zredukowane do tagów div.